This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) An embedded system configured to reduce volatile memory usage by loading individual software components, the embedded system comprising:

a processor;

volatile memory in electronic communication with the processor;

non-volatile memory in electronic communication with the processor, the non-volatile memory including an operating system, a loader application, a loading table that is configurable by a user, and a plurality of individual software components;

instructions stored in the non-volatile memory that are executable by the processor for implementing a method comprising:

loading the operating system for the embedded system into the volatile memory;

starting the operating system;

loading the loader application into the volatile memory;

starting the loader application;

examining the loading table to determine which of the individual software components are to be loaded into the volatile memory; and

loading each of the individual software components that are to be loaded as indicated in the loading table into the volatile memory.

2. (Original) The embedded system as defined in claim 1, wherein the embedded system is a multi-functional peripheral.

3. (Canceled)

4. (Currently Amended) The embedded system as defined in claim 1~~3~~, further comprising an input component in electronic communication with the processor for a user to enter user input and thereby configure the loading table.

5. (Original) The embedded system as defined in claim 4, further comprising a display in electronic communication with the processor that displays information to the user relating to the loading table.

6. (Original) The embedded system as defined in claim 5, further configured with a menu structure that may be navigated by a user using the input component and the display to configure the loading table.

7. (Original) The embedded system as defined in claim 6, wherein the loading table is directly configurable by a user.

8. (Original) The embedded system as defined in claim 6, wherein the loading table is indirectly configurable by a user.

9. (Original) The embedded system as defined in claim 1, wherein the loading table is a license table comprising a list of licenses relating to the individual software components.

10. (Original) The embedded system as defined in claim 9, wherein the individual software components with licenses, as indicated by the license table, are loaded into the volatile memory.

11. (Original) The embedded system as defined in claim 1, wherein the volatile memory is RAM.

12. (Original) The embedded system as defined in claim 1, wherein the individual software components are software libraries.

13. (Original) The embedded system as defined in claim 1, further comprising:

a communications module in electronic communication with the processor for communications with a computer; and

a web interface accessible by a user through use of a web browser to configure the loading table.

14. (Original) The embedded system as defined in claim 13, wherein the web interface comprises a web page.

15. (Original) The embedded system as defined in claim 1, wherein the method further comprises:

examining hardware configuration by the loader application; and

modifying the loading table based on the hardware configuration.

16. (Currently Amended) A computer-readable medium for carrying program data, wherein the program data comprises executable instructions for implementing a method comprising:

loading an operating system for an embedded system into volatile memory;

starting the operating system;

loading a loader application into the volatile memory;

starting the loader application;

examining a loading table that is configurable by a user to determine which individual software components are to be loaded into the volatile memory; and

loading each of the individual software components that are to be loaded as indicated in the loading table into the volatile memory.

17. (Original) The computer-readable medium as defined in claim 16, wherein the embedded system is a multi-functional peripheral.

18. (Original) The computer-readable medium as defined in claim 16, further comprising a user configuring the loading table.

19. (Original) The computer-readable medium as defined in claim 18, further comprising providing a user interface to the user for configuring the loading table.

20. (Original) The computer-readable medium as defined in claim 19, wherein the user interface includes a menu structure that may be navigated by the user to configure the loading table.

21. (Original) The computer-readable medium as defined in claim 20, wherein the user configures the loading table directly.

22. (Original) The computer-readable medium as defined in claim 20, wherein the user configures the loading table indirectly.

23. (Original) The computer-readable medium as defined in claim 16, wherein the loading table is a license table comprising a list of licenses relating to the individual software components.

24. (Original) The computer-readable medium as defined in claim 23, wherein the individual software components with licenses, as indicated by the license table, are loaded into the volatile memory.

25. (Original) The computer-readable medium as defined in claim 16, wherein the volatile memory is RAM.

26. (Original) The computer-readable medium as defined in claim 16, wherein the individual software components are software libraries.

27. (Original) The computer-readable medium as defined in claim 16, further comprising providing a web interface accessible by a user through use of a web browser to configure the loading table.

28. (Original) The computer-readable medium as defined in claim 27, wherein the web interface comprises a web page.

29. (Original) The computer-readable medium as defined in claim 16, further comprising:
examining hardware configuration by the loader application; and
modifying the loading table based on the hardware configuration.

30. (Currently Amended) A method for reducing volatile memory usage in an embedded system by loading individual software components, the method comprising:

loading an operating system for the embedded system into volatile memory;

starting the operating system;

loading a loader application into the volatile memory;

starting the loader application;

examining a loading table that is configurable by a user to determine which individual software components are to be loaded into the volatile memory; and

loading each of the individual software components that are to be loaded as indicated in the loading table into the volatile memory.

31. (Original) The method as defined in claim 30, wherein the embedded system is a multi-functional peripheral.

32. (Canceled)

33. (Currently Amended) The method as defined in claim 30~~32~~, further comprising providing a user interface to the user for configuring the loading table.

34. (Original) The method as defined in claim 33, wherein the user interface includes a menu structure that may be navigated by the user to configure the loading table.

35. (Original) The method as defined in claim 34, wherein the loading table is directly configurable by a user.

36. (Original) The method as defined in claim 35, wherein the loading table is indirectly configurable by a user.

37. (Original) The method as defined in claim 30, wherein the loading table is a license table comprising a list of licenses relating to the individual software components.

38. (Original) The method as defined in claim 37, wherein the individual software components with licenses, as indicated by the license table, are loaded into the volatile memory.

39. (Original) The method as defined in claim 30, wherein the volatile memory is RAM.

40. (Original) The method as defined in claim 30, wherein the individual software components are software libraries.

41. (Original) The method as defined in claim 30, further comprising providing a web interface accessible by a user through use of a web browser to configure the loading table.

42. (Original) The method as defined in claim 41, wherein the web interface comprises a web page.

43. (Original) The method as defined in claim 30, further comprising:
examining hardware configuration by the loader application; and
modifying the loading table based on the hardware configuration.